Evaluation of a Semi-Automated Semantic Annotation Approach for Bootstrapping the Analysis of Large-scale Web Service Networks

Shahab Mokarizadeh[1], Peep Küngas[2], Mihhail Matskin[3]

[1] *Royal Institute of Technology (KTH), Stockholm, Sweden*
[2]*University of Tartu (UT), Tartu, Estonia*
[3] *Norwegian University of Science and Technology (NTNU), Trondheim, Norway*
[1]shahabm@kth.se, [2]peep.kungas@ut.ee,[3]misha@kth.se

***Abstract*—In recent years many methods have been proposed, which require semantic annotations of Web services as an input. Such methods include discovery, match-making, composition and execution of Web services in dynamic settings, just to mention few. At the same time automated Web service annotation approaches have been proposed for supporting application of former methods in settings where it is not feasible to provide the annotations manually. However, lack of effective automated evaluation frameworks has seriously limited proper evaluation of the constructed annotations in practical settings where the overall annotation quality of millions of Web services needs to be evaluated. This paper describes an evaluation framework for measuring the quality of semantic annotations of large number of Web services descriptions provided in form of WSDL and XSD documents. The evaluation framework is based on analyzing network properties, namely scale-free and small-world properties, of Web service networks, which in turn have been constructed from semantic annotations of Web services. The evaluation approach is demonstrated through evaluation of a semi-automated annotation approach, which was applied to a set of publicly available WSDL documents describing altogether ca 200 000 Web service operations.**

Keywords--Web service annotation; Web service network; WSDL; XSD

I. INTRODUCTION

Many methods have been recently proposed in the field of Web services, which exploit knowledge-rich descriptions of services for various purposes such as discovery, match-making, composition and invocation. In addition, there has been recently some activity [2][3][4][5] in web service discovery and composition for tackling the analysis of essential properties of Web services networks with the aim to provide feedback to Web services discovery and composition problems by revealing the essential characteristics of search spaces of particular problems. These knowledge-based methods usually assume existence of semantic annotations of Web services. The annotations, however, are not available in general and this has motivated research on methods for automated Web service annotations. Although great progress has been made in this thread of research, lack of evaluation methodologies suitable for evaluating automatically annotations of thousands or even millions of Web services has hindered proper evaluation of the proposed solutions and their results in settings where manual evaluation is not feasible. The latter in turn has delayed application of the annotation methods for generating high-quality input, which will enable evaluation or application of knowledge-intensive methods in large-scale settings.

More specifically, the work in Web service network analysis suffers from three major drawbacks. First, due to lack of semantic annotations in the vast majority of existing web services only syntactic matching is exploited for analysis [2]. Second, only small sets of semantically annotated web services are examined [4][5] due to the high costs related to manual labor required for annotation of web services [7]. Third, the methods rely on the assumption that the supporting reference ontology is provided [3]. Since semantic annotations enable construction of dataflow- and workflow-based networks, which are needed for this kind of analysis, and considering the outlined deficiencies there is a need for a (semi-)automatic cost-effective web service annotation mechanism, which has been evaluated with respect to large amount of Web service descriptions.

In this paper, we propose an evaluation framework, which is suitable for effective evaluation of large-scale annotation efforts. The framework is applied to a novel semi-automatic cost-effective semantic annotation method, which is introduced as well. The latter combines a previously proposed ontology learning method [6] and a cost-effective semi-automatic semantic annotation method [1], for bootstrapping analysis of large repositories of Web services currently without semantic annotations. The annotation method starts with semi-automatically generating reference ontology. The latter is then exploited to annotate automatically the Web services interfaces (WSDL documents) under examination.

After that annotation method is applied we evaluate it by using the proposed framework. The first step of the evaluation framework measures performance of annotations with respect to the generated reference ontology and the respective golden ontology. In order to provide an unbiased assessment, the evaluation is performed over two different small datasets. The second step incorporates much larger subsets into evaluation, and examines the characteristics of the semantic networks formed by automatically annotated web services. Based on these assessments, we provide analytical metric to track the performance of the annotation method with respect to specific

network properties. These properties will indicate relative quality of the annotations.

The rest of this paper is organized as follows. In Section 2, we outline our web service annotation method. In Section 3, we discuss the framework to evaluate the quality and quantity of applied annotations. Experimental results are presented in Section 4. Finally, Section 5 reviews related work, while conclusions and future work are presented in Section 6.

II. AUTOMATED WEB SERVICE ANNOTATION

Our automated annotation method combines a previously proposed ontology learning method [6] and a semi-automatic semantic annotation method [1] whereas the former provides input to the latter in form of annotation heuristics. The learning method is used to generate reference ontology from a corpus of Web service descriptions and then utilize the generated reference ontology to generate annotation heuristics. In the reference ontology, instances are referring to the *terms* while classes refer to conceptual representation of the underlying *terms*. In the context of this paper, *term* refers to an XML schema basic element name or a WSDL message part name in the corpus of Web services.

The learning method uses Bag-of-Words model [20] (where any relation between terms are ignored) for retrieving the terms from WSDL documents. The extracted terms build up the dataset which is considered for annotation. Moreover, the conceptual classes in the generated ontology are linked through ontological properties such as *hasProperty* and *isSynonymOf*. While *isSynonymOf* conveys that the associated concepts are lexically synonyms, *hasProperty* relation expresses other kind of relationship between concepts (e.g. "Person hasProperty Name" states that concept *Person* has property conceptualized as concept *Name*).

In order to learn ontologies for matching inputs and outputs of web services through annotated elements, we rely on a set of matching rules. Here matching refers to the process of finding relationship (i.e. correspondence) between instances (*terms*) in an ontology through utilization of any of following rules. In other words, we examine whether the instances belong to same *synset* or not. A *synset* in the context of this paper is a group of *terms* that are considered semantically equivalent. We consider two instances matched if and only if one of the following conditions is true:

Rule-1: They both belong to a same concept (e.g. *{loc, location1} instanceOf Location*).

Rule-2: They belong to lexically synonym concepts (e.g. *loc instanceOf Location* and *place instanceOf Place,* where *Place isSynonymOf Location).*

Rule-3: One of the instances belongs to a concept which subsumes the concept representing the second instance (e.g. pair of *{ConrtactId, Id}* where *ContractId instanceOf ContractIdentifier, ContractIdentifier isSubClassOf Indentifer* and *id instanceOf Identifier).*

Rule-4: One of the instances belongs to a synonym concept which subsumes the concept representing the second instance (e.g. pair of {*bidUId, Id*} where *bidUId instanceOf BidUniqueCode, BidUniqueCode isSynonymOf ContractIdentifier* and *id instanceOf ContractIdentifier).*

Rule-5: The instances belong to two concepts that are inter-related by an ontological property other than *isSynonymOf, isSubcalssOf and isSuperClassOf* (e.g. *Person hasProperty FirstName*).

The heuristic-based annotation mechanism accepts annotation heuristics represented as rules in form: *entity_reference* ← *synset* (e.g. *Password* ← *{password, pwd, strPassword, authpassword, pass*}). The meaning of such a rule is that an XML schema element matched by any element in the *synset* is annotated with the entity reference (in our case a concept identifier in the automatically constructed ontology). We construct *synsets* from the labels of particular instances of the reference ontology. Thus according to the previous heuristic rule example – if *Password* is a concept identifier, then *password, pwd, strPassword, authpassword* and *pass* will be terms.

By utilizing the generated ontology and annotating respective web service elements, we promote the process of correlating web service inputs and outputs from pure syntactic level to ontological instance matching level. The annotation method annotates those extracted schema element names / message part names (*terms*) with their respective concepts in the generated ontology.

III. EVALUATION APPROACH

Our evaluation framework performs the following steps to evaluate the proposed annotation method:

1) **Evaluation of the web service annotation and matching scheme.** We employ our annotation scheme to annotate certain elements (*terms*) from web service corpus by generating a reference ontology. Next, we evaluate quality and quantity of matching cases discovered using our introduced matching rules. In addition, we are looking at the annotation progress for a large repository of Web service corpora.
2) **Analysis of the generated web service networks.** We construct a Web service network out of those annotated Web services, then investigate models, governing the network, and compare it with previous observations [2][3][4][5]. Finally we measure the effectiveness of the adopted matching scheme and preceding ontology learning mechanism with respect to network properties of the Web service network.

A. Web service Matching Evaluation Method

In order to provide an unbiased evaluation of the generated reference ontology and subsequent annotations, we perform the assessment over two different datasets. The first dataset embodies 2000 most frequent element names taken from our collection of WSDL corpora[1](ca. 15000 WSDL documents collected from different repositories in the Web). The second dataset incorporates all harvested terms (with any frequency) from the collection of 146 WSDL documents[2] which were

[1]Available at : http://www.soatrader.com/web-services

[2]Available at : http://www.andreas-hess.info/projects/annotator/

annotated by ASSAM tool by Hess et al. [12]. This category includes 375 unique terms. From now, we refer to the first and second dataset as Top2000 and ASSAM respectively and we process both datasets in a same way. For annotation of services in both datasets, we create two independent reference domain ontologies. While the first domain ontology is constructed manually by an ontology engineer, the second one is constructed automatically using our ontology learning mechanism [6]. We refer to the former and the latter respectively as the golden and the generated ontology.

For Top2000, the golden ontology is handcrafted by authors, while in case of ASSAM we use the ontology developed by Hess et al. [12] and exploited as reference ontology in their experiment with ASSAM tool [12]. The main difference between these two golden ontologies is that, unlike Top2000, concepts in the golden ontology of ASSAM dataset are also inter-related by property relations which were derived from structural relationship between message and part-names elements in their collection of WSDL documents. We acknowledge that at least the golden ontology of Top2000 category might suffer from bias introduced by the human expert mainly due to the lack of documentation in underlying Web service corpus .As the automatic Web service matching is the target use-case for annotated web services, we investigate the quality of pair-wise semantic matches between annotated XML schema element/ part names (i.e. ontological instances). The quality is measured in terms of precision, recall and F-measure of matched instances using automatically generated ontology compared with those in the golden counterpart.

In the following, terminology / formalism of Euzenat and Shvaiko [11] is used to describe our instance matching process. The result of instance matching process is a set of correspondence elements. Each correspondence element implies that a relation holds, according to a particular matching rule, between two instances in an ontology. A correspondence element $Ont_k C_{i,j}$ is a 3-tuple $<a_i, b_j, R>$ where $i,j=1...N, i \neq j$; N is the number of instances; a_i and b_j refer to *i-th* and *j-th* instances in the ontology referenced by Ont_k; where k is the identifier of the ontology, and finally R specifies the matching rule that reveals kind of semantic relationship holding between a_i and b_j. If the instances are not matched, then we use notation of *NM (NotMatched)* instead of the matching rule. For evaluation purpose of each category of datasets, we compare the matching rules R and R' in $Ont_{Gen}C_{i,j}=<a_i, b_j, R>$ and $Ont_{Gold}C'_{i,j}=<a_i, b_j, R'>$ where $Ont_{Gen}C_{i,j}$ denotes the correspondence element obtained in the generated ontology (Ont_{Gen}) while $Ont_{Gold}C'_{i,j}$ refers to the computed correspondence element for the same pair of instances a_i and b_j in the golden ontology(Ont_{Gold}).

B. Web Service Network Evaluation Method

It should be noted that the automatically generated reference ontology is not ideal yet, hence; we need to enhance its quality by incorporating for example other external resources such as domain ontologies or by exploiting structured data model instead of flat bag-of-words model. Moreover, as the size of web service corpus increases, (semi-) automatically verifying the generated reference ontology for instance by aligning it with other ontologies or human expert intervention will be challenging. Hence, it is not cost-effective to incorporate the entire dataset for ontology learning purpose. Instead, we aim to discover and annotate an optimum subset of entire dataset which its network characteristics exhibit closest approximation to the already observed properties in smaller Web service networks. The result of this network evaluation provides feed-back to the employed Web service matching and annotation scheme. In our first attempt, to discover this ideal dataset, we perform experiments with datasets resulted from applying four different thresholds where each threshold represents a minimum occurrence frequency of terms in the entire dataset of WSDL/XSD elements (approx. 1,000,000 terms) from our collection of WSDL documents. We extract four datasets using four (arbitrary chosen) thresholds 10, 15, 20 and 25, exploit them first for ontology learning and annotation and finally Web service network formation. We refer to corresponding datasets by *h10, h15, h20, h25* and they cover altogether around top 30000 most recurrent terms.

In order to make our results comparable with related work, we present our Web service network models based on similar principals proposed by [2][4][5]. We model a web service semantic network by a 2-tuple model *M=(T, O)*, where:

- *T* : is the type of nodes in the network model and can be either web service (*ws*), operation (*op*) or parameter part name or XML schema element name (*p*) , abstracted by a concept in the ontology *O*.
- *O*: is the identifier of the exploited ontology for web service matching purpose.

As an illustrative example of a semantic parameter network, consider Fig. 1 where on the left hand side of the figure a web service network is formed by a set of web services (WS_1, WS_2, and WS_3), each of which consists of one operation (OP_1,OP_2, and OP_3 respectively). While P_1 - P_6 are expressing message part name or XML schema element name instances of input/output parameters of their respective operations, the ontological concepts representing these parameter instances are symbolized by C_1 – C_5. The right hand side of Fig.1 denotes the resulting semantic parameter network which is a transformation of the left side network, where the parameter instances are replaced with respective

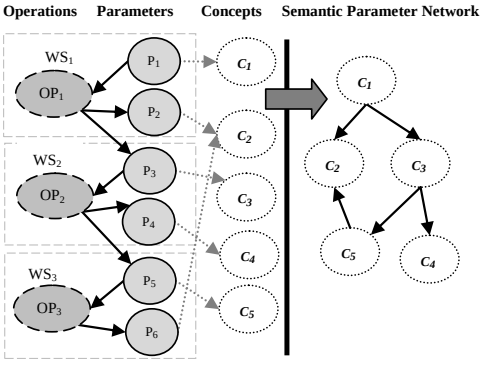

Figure 1. Example of Semantic Parameter Network Formation.

concepts and operation nodes are transformed to directed edges from input concepts to the respective output concepts.

Topological landscape of Web service networks formed by real world dataset has shown [2][4][5] to exhibit characteristics of both small-world [8] and scale-free [10] networks at least for small networks of Web services. This implies that networks constructed out of annotated Web services are complex networks exhibiting the following properties: *i) power-law degree distribution ii) small average path-length iii) high clustering coefficient*. Degree distribution refers to the probability distribution by which number of edges for a given random vertex in the network can be computed. Average-path length is the distance between two vertices averaged over all pairs of nodes, while clustering coefficient is the average fraction of pairs of neighbors of a vertex that are also neighbors of each other. In addition, we are examining *correlation degree* of the networks as an indication of emergence of social-network properties [13]. Accordingly, a network is said to have positive correlation (aka. *assortative mixing*) on its degree if vertices with high number of connection tend to be connected with other nodes which also have many links. Alternatively, if the preference is to attach to those having small quantity of connection, then it is said to have negative correlation on degrees (aka. *dis-assortative mixing*). A recent study by Newman [13] has shown that technological and biological networks (e.g. Internet, WWW, protein interactions) are exhibiting negative correlation on degrees whereas positive correlation is mainly observed in social networks (e.g. network of actors).

In our network evaluation, we initially investigate scale-free and small-world characteristics of constructed networks and compare results with previous findings. The emergence of these properties supports the validity of our annotation and matching scheme as they preserve the previously recorded characteristics of the service networks. Next, we propose our metrics to analytically evaluate the performance of the adopted matching scheme in context of network metrics. Such analytical metrics can be considered as a mean to track the effectiveness of adopted matching scheme and preceding ontology learning mechanism in web service semantic annotation paradigm.

IV. Evaluation Results

In this section, we are presenting the experiments[3] we made over all three datasets (Top2000, ASSAM, and the entire collection) to generate reference ontology, annotate web service elements and to evaluate Web service networks.

A. Matching Results

Our ontology learning system cannot generate annotation heuristics covering all terms in the given datasets; hence, the number of instances in the generated ontologies can be smaller than those in the golden counterpart. For the matching purpose, we are only taking into account the common instances between golden and generated ontologies and they account for 1600 (out of 2000) and 254 (out of 375) cases for Top2000 and ASSAM datasets respectively. In fact these common instances denote the number of *terms* successfully processed and end up with an ontological concept; hence, they can be considers as processing recall of our ontology learning system. Evaluation of matching results is based on comparison between R to R' in $Ont_{Gold}C'_{i,j}=\langle a_i, b_j, R'\rangle$ *and* $Ont_{Gen}C_{i,j}=\langle a_i, b_j, R\rangle$ for all pairs of correspondence elements, as already pointed out in Section 3.1. The resulting correspondence element pairs are grouped into three disjunctive sets: true positives (*TP*) (the correspondence elements which are common between golden and generated ontology), false positives (*FP*) (correspondence elements discovered only by generated ontology) and false negatives (*FN*) (those which are matched by the golden ontology but not discovered by the generated ontology). Based on these three groups, Precision (P), Recall (R) and F-measure (F) are computed for different matching rules for both datasets, using following formulas:

$$P = \frac{TP}{TP+FP},\ R = \frac{TP}{TP+FN},\ F = \frac{2*P*R}{P+R} \qquad (1)$$

The computed values for the aforementioned metrics for both datasets are further detailed based on the utilized group of matching rules namely: Rule 1, Rules 1-4 and Rules 1-5 (all rules) and they are shown in Table 1. The utilized matching rule(s) are basically expressing which rules are exploited for identifying the correspondence elements.

According to Table 1, precision of matching is smoothly decreasing as more loosely coupling matching rules (namely Rules 2-5) are incorporated, while at the same time recall is increasing sharply from Rule-1 to Rules 1-4 and slightly from Rules 1-4 to Rules 1-5 for both datasets. This trend reveals the fact that although incremental combination of matching rules is not enhancing the accuracy; the proportion of matched instances is increasing significantly. In other words, when a given part/element name is assigned to an imprecise concept, then designated matching rules can not compensate this deficiency. In our work, accuracy of matching is mostly affected by performance of our ontology learning steps [6] which in turn, partially depends on the syntactic (e.g. spelling,…) and semantic (e.g. ambiguity) quality of part/element names and designated lexico-syntactic patterns. Regarding ASSAM dataset, it can be clearly seen in Table 1 that recall values for all three groups of matching rules is considerably smaller than those for Top2000. This phenomenon is partially associated to the ambiguity of given names and the lexico-syntactic patterns exploited in ontology learning steps which are designed based on frequent non-

TABLE I. Precision, Recall an F-measure of Identified Correspondence Elements.

		Rule-1	Rules 1-4	Rules 1-5
Top2000	**P**	0.315	0.2007	0.2008
	R	0.1961	0.3836	0.4275
	F	0.241	0.2635	0.2732
ASSAM	**P**	0.495	0.447	0.3787
	R	0.046	0.1043	0.1068
	F	0.0841	0.1691	0.1666

[3]Experiments available at: www.isk.kth.se/~shahabm/WSAnalysis/

ambiguous syntactic patterns observed in our dataset. Since ASSAM dataset is very small and therefore embodies small proportion of those frequent patterns, it is reasonable to observe relatively lower recall values. The higher precision of ASSAM dataset compared to the Top2000 one is partially related to the smaller size of processed names in ASSAM group, which diminishes the effect of false positives. In overall, observing the increasing trend for F-measure from single matching rule (Rule-1) toward full set of matching rules (Rules 1-5) reveals the fact that exploitation of designated matching rules improves the matching performance. From quite modest point of view and by ignoring the linguistic ambiguity of dataset, one can generalize ASSAM metrics as rough estimations (i.e. lower-bound) of achievable performance by annotation of our entire WSDL collection. Similarly, Top2000 measures can be interpreted as the best figures we can achieve as it covers the top recurrent nominees.

It should be noted that the main objective of exploiting bag-of-words model is to provide a baseline for our analysis and a comparison base for further improvements rather than pushing forward the state of art techniques in this field. We acknowledge that despite of the novelty of the matching rules, they are incapable to cover major identified relations[14] in a compound noun. For example, the adopted lexico-syntactic patterns and matching rules only take into account *Be* and *Have* relations between head and modifier parts of a compound noun and other relations such as *About, Instance, In* and, *Actor* are ignored. Identifying these semantic relations will also enrich the generated ontology and improves both precision and recall of ontology learning and subsequent instance matching. Moreover, the evident shortcoming of bag–of-words model, which ignores relations between elements, leads to scattered structure of the generated ontology, which affects negatively usage of Rule-3.

B. Web Service Annotation Progress

The experimental results for annotation coverage of top 30000 most frequently occurring terms are depicted in Table 2. Accordingly the first row shows the size of *h10, h15, h20* and *h25* datasets (only unique elements are counted) where third and fourth rows reveal how many overall annotations could be provided respectively with thresholds of 10, 15, 20 and 25. Total number of elements to be annotated is denoted in the second row. According to Table 2, we can annotate 59% of WSDL part names or XSD type fields just by applying the ontology generated from dataset with occurrence frequency of 25. Moreover, while the volume of dataset with frequency 10 is over two times larger than those with frequency 25, it enhances the annotation progress only by 7%. It can be seen that despite the substantial increase in size of ontology learning inputs (i.e. those four thresholds), the quantity of resulting annotations shows a slight growth. In other words, more and more ontological concepts need to be generated in order to annotate the less recurrent elements.

If we compare the statistics in Table 2 and the percentage of achieved annotations from possible ones with the results presented by Küngas and Dumas [1] on their semi-automatic cost-effective approach on a smaller case study, we can see that the automated ontology learning technique allows to achieve similar annotation coverage to a human expert, after annotation heuristics are constructed automatically from concept instances. Thus one of the contributions of our presented ontology learning approach is to reduce the number of man-hours required in a cost-effective annotation scheme even further while still providing the same coverage.

TABLE II. GENERAL STATISTICS OF ANNOTATION PROGRESS.

	h25	**h20**	**h15**	**h10**
Learned ontology size	4523	5614	7378	11610
Annotated elements	588057	596625	621336	663618
Total elements	998916	998916	998916	998916
Percentage of total	59%	60%	62%	66%

C. Constructed Web Service Networks

Due to the space restriction, we are limiting ourselves to the analysis of semantic parameter networks ($T=p$) using ontologies generated from datasets *h10, h15, h20, h25*, the golden ontology and its counterpart automatically generated ontology of both Top2000 and ASSAM datasets (i.e. O_{h10}, O_{h15}, O_{h20}, O_{h25}, $O_{Gold\_Top2000}$, $O_{Gen\_Top2000}$, O_{Gold_ASSAM} and O_{Gen_ASSAM} respectively). Moreover, in order to obtain the baseline for different matching scheme, all the generated networks are constructed only using the first matching rule (i.e. Rule-1) unless it is explicitly specified. Aiming to provide evidence, which will support our hypothesis, we also compare them with counterpart networks constructed based on pure syntactic matching, random annotation matching (where ontological concepts are randomly assigned to elements) and also classic random complex networks. Similar to many studies on the small world networks [8], the analysis is restricted to the giant components in the networks (i.e. the maximal connected sub-graph of the network). Our hypothesis is that the networks constructed based on an acceptable matching threshold preserve characteristics of both scale-free and small-world networks and exhibits negative correlation degree, similar to already observed properties in smaller Web service networks.

Analysis of Small-World Properties. According to Watts and Storgatz [8], small world networks are networks with the following characteristics: 1) exhibiting small average shortest path length, and 2) exposing high clustering coefficient. These properties are measured by the average shortest path and clustering coefficient metrics, which are denoted by L and C symbols respectively in Table 3. In the interest of verifying small-world characteristic, the computed metrics in the target networks are compared with those estimated from similar random network generated based on *Erdos&Renyi* (ER) model [9] (with same number of nodes and edges appearing in the actual network). The computed average shortest path and average clustering coefficient metrics for the random network are denoted by L_{Random} and C_{Random} symbols and for the actual parameter network by L_{Actual} and C_{Actual} symbols respectively. If a network exposes the small world properties, then it is expected that $L_{Random} \lesssim L_{Actual}$ (i.e. average shortest path is

TABLE III. SCALE-FREE AND SMALL-WORLD PROEPRTIES OF EXAMINED NETWORKS. **L**:AVERRAGE PATH LENGTH**, C**: CLUSTERING COEFFICEINT, $\mathbf{S_{INDEX}}$:INDEX OF SMALL-WORD-NESS.

	Networks		L	C	S_{index}
Entire	Syntactic	**Actual**	**3.283**	**0.2968**	**591.08**
		Random-ER	*3.9229*	*0.00062*	
h 25	Generated	**Actual**	**2.4256**	**0.2590**	**7.5769**
		Random-ER	*2.4756*	*0.0348*	
h20	Generated	**Actual**	**2.3882**	**0.2811**	**8.8148**
		Random-ER	*2.4851*	*0.0331*	
h15	Generated	**Actual**	**2.3724**	**0.2805**	**8.2753**
		Random-ER	*2.3396*	*0.0334*	
h10	Generated	**Actual**	**2.5322**	**0.2449**	**18.2709**
		Random-ER	*2.7662*	*0.0146*	
Top2000	Golden	**Actual**	**2.1895**	**0.3761**	**2.8404**
		Random-ER	1.8852	0.1146	
	Generated	**Actual**	**2.08475**	**0.3209**	**3.3878**
		Random-ER	*2.0667*	*0.0939*	
ASSAM	Golden	**Actual**	**4.5653**	**0.2147**	**3.1464**
		Random-ER	*3.5460*	*0.05304*	
	Generated Rule. 1	**Actual**	**3.0592**	**0.4803**	**21.4835**
		Random-ER	*3.8451*	*0.0281*	
	Generated Rules .1-4	**Actual**	**2.5732**	**0.4057**	**8.5288**
		Random-ER	*3.1267*	*0.0578*	

almost equal or slightly larger than of a random network) and $C_{Actual} \gg C_{Random}$ (i.e. average clustering coefficient is much larger than that of a random network) [8]. In order to explore the extent to which the small-world topology changes with parameter variation, we exploit a measurement of 'small-worldness', shown by S_{index} ,proposed by Humphries and Gurney[15] which is defined as :

$$\gamma=\frac{C_{Actual}}{C_{Random}},\ \lambda=\frac{L_{Actual}}{L_{Random}}\ , S_{index}=\frac{\gamma}{\lambda} \qquad (2)$$

According to [8], in order to meet small world criteria given above, the network model should fulfill the following conditions: $\gamma >> 1$, $\lambda > 1$, and $S_{index} > 1$. The S_{index} scales linearly with the size of vertices of the network. We use S_{index} metric to compare networks constructed using different matching scheme and dataset with respect to their small-world properties.

The computed small world network metrics for both actual and counterpart random networks (marked by "Actual" and "Random-ER" suffixes) are shown under columns *L* and *C* of Table 3. It can be seen that in all networks both small world conditions, $L_{Random} \lesssim L_{Actual}$ and $C_{Actual} \gg C_{Random}$ are holding, despite to the fact that each network exhibits a different level of small world properties. However, one can also observe a slight increasing trend in violating form the first condition of small-worldness as we are moving from small size networks (starting form *h25* generated network with 2086 nodes) toward larger networks (*h10* with 4050 nodes and pure syntactic network with 67622 vertices). While holding only the second condition for small networks (200–3000 vertices) is sufficient to demonstrate small-world properties[16], occurrence of significant deviation in average shortest path length, especially for large networks, is alarming. We suggest that an efficient matching scheme should eliminate (or at least minimize) such a violation, which is mostly due to the fact that adding new concepts to the Web service network is not growing at the same rate as annotations of new web service operations (i.e. adding new edges). This phenomenon is the consequence of two features. First, our annotation method start with most recurrent elements of Web services and therefore annotating a new Web service operation is more likely to introduce a new edge (thus increasing clustering coefficient) rather than adding a new node, hence not increasing the path length. Second reason is the deficiency in ontology learning and annotation to identifying the target concepts correctly. The effect of former deficiency can be tracked in the fluctuation of small-world index of *h25* toward *h10*, which is mostly the result of the change in shortest length path. We will look at the effect of annotation scheme in network properties in the end of this section.

Analysis of Scale-free Properties. Emergence of power-law degree distribution in a network, as a prominent sign of scale free networks, implies that few vertices are highly connected, whereas majority of vertices have a low degree of connectivity (i.e. existence of hub nodes) [10]. Expecting to observe the same pattern, we examine the outgoing edge distribution in the generated networks. The results for all six categories of networks are fitted to a power law function $y = cx^{-\alpha}$ (in log-log plot) where x represents the outgoing degree and y denotes the frequency of nodes with the same outgoing edge degree. Due to the space restriction, we only show the plot for the parameter network of *h15* in Fig. 2 (as sample representative of other plotted distributions) and summarize the exponents of power-law function (i.e. α) for all networks in the first column of Table 4. Plotting the network in log-log scale shows that all networks are presenting near power-law like distribution with the exponent ranging from 1.1448 to 1.5316. By comparing these values to the exponent of 1.3903 (the result which Oh et al. [2] reported with pure syntactic matching scheme for a network with 4456 nodes and diameter of 8), we could see that we get a smaller power- law exponent. This is mainly due to the difference in the exploited Web service matching scheme (syntactic vs. semantic). In fact, a semantic network embodies smaller number of nodes than its syntactic counterpart although exposes higher (in/out) degree because each node in the semantic network represents (semantically annotated) at least one counterpart syntactic node. In order to verify that such emergence of power-law distribution is not a random phenomenon, we selected two networks (*h25* and golden ASSAM) as representative of small and medium size graphs from two different datasets and created their counterpart networks based on random annotations of given Web service element/part names. These

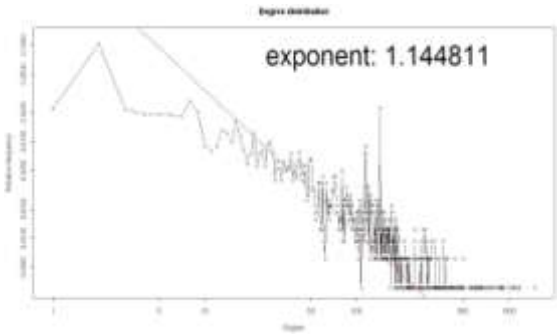

Figure 2. Out-degree distribution plot for generated network of h15.

networks are highlighted with "Random Annotation" affixes in Table 4. Plotting these networks reveals that they exhibit normal (Gaussian) distribution rather than expected power-law distribution.

Analysis of Effect of Matching Scheme Performance on Network Properties. In this section we summarize the lessons learned from analysis of network properties with regards to performance of exploited matching and annotation scheme. The goal is to provide guideline metrics to estimate the accuracy of resulting networks and also provide proper feedback to exploited matching and annotation components.

1. *Small-worldness*. A comparison between small world metrics of golden and generated networks for both ASSAM and Top2000 reveals the influence of efficiency of adopted matching scheme over small-world index. If we consider F-measure (depicted in Table 2) as proper indication of matching scheme performance, it can be seen in Table 3 that the generated ASSAM network using Rules 1-4 (F= 0.16913) is exhibiting much closer approximation to S_{index} of counterpart golden network than the one generated using only Rule-1 (F=0.0841). Similarly, we can see that deviation between S_{index} of Top2000 golden and generated networks (2.8404 vs. 3.3878) is much less than observed difference in counterpart ASSAM networks (3.1464 vs. 8.5288). This is also due to difference in the efficiency of utilized matching schemes (F=0.241 for Top2000 vs. F=0.1691 for ASSAM). This observation together with the finding of Humphries and Gurney[15], that small-worldness scales linearly with network size, supports the hypothesis that in ideal matching scheme over incremental set of data, it is expected to observe a harmony in the growth of clustering coefficient and shortest length path, as network expands. Hence we need to see a linear growth from the index of Top2000 golden network (as the lower bound index for *h** networks) toward *h10* and emergence of any significant fluctuation is more probably revealing an alarming point.
2. *Degree of power-law exponent.* Our hypothesis is that, in an acceptable threshold of matching scheme performance, the generated network exhibits still power-law distribution where its exponent lies lower than that of syntactic network and keeps distance from normal distribution exposed by counterpart randomly annotated networks. The lower bound comes from the worst case assumption, when each part name/schema element name refers to a unique ontological concept; hence there will be no difference between syntactic and semantic networks. On the other hand, the upper bound threshold is achieved when the degree distribution of a generated network is not fitted well to power-law degree distribution and it rather tends to fit normal degree distribution (bell shaped), which implies the absence of expected hub nodes in the network. Clauset et al. [17] formalized practical methods to recognize a power-law distribution from other kind of distributions for a given graph. Their findings can be exploited for tracking deviation from power-law distribution.

TABLE IV. SCALE-FREE PROPERTIES OF NETWORKS **P**: POWER-LAW DEGREE EXPONENT, **N**: NUMBER OF NODES IN THE NETWORK, **D**: DEGREE CORRELATION.

Category	Networks	P	N	D
Entire	Syntactic	1.3722	67622	-0.0413
h 25	Generated	1.1945	2086	-0.1993
	Random Annotation	*0.6332*	*2086*	0.0190
h20	Generated	1.1977	2394	-0.2093
h15	Generated	1.1448	3239	-0.2222
h10	Generated	1.2316	4050	-0.1895
Top2000	Golden	1.1504	856	-0.2238
	Generated	1.1483	936	-0.2137
	Syntactic	1.1653	828	-0.2229
ASSAM	Golden	1.5346	170	-0.3079
	Generated- Rule. 1	1.5574	413	0.3642
	Generated - Rules .1-4	1.4566	217	0.0410
	Random Annotation	1.0755	170	0.1151
	Syntactic	1.6105	886	0.1940

3. *Correlation degree on nodes*: It can be seen in Table 4, that all three pure syntactic networks, entire (constructed from our whole dataset), ASSAM and Top2000 are exposing negative correlation (disassortative mixing) on their degrees. Emergence of this behavior supports the seminal finding that Web service networks, similar to many other technological networks [13], exhibit disassortative mixing on degrees. As it is was expected, all the generated semantic networks, except ASSAM, preserve the nature of their syntactic origin and show negative correlation degree. This behavior of ASSAM network is due to the deficiency of exploited matching scheme for this network which fails to annotate sufficient quantity of Web services correctly such that the resulting semantic network could exhibit its intrinsic properties. In order to illustrate this deficiency, we constructed ASSAM network based on matching Rules 1-4, which expose higher F-measure than current Rule-1. According to Table 4, moving from F-measure of 0.0841 to 0.1691 results in significant enhancement (toward expected value) in the correlation degree and shifting it from absolute positive value to almost zero. Unlike ASSAM generated networks, the correlation degree of Top2000 generated network exposes much better approximation to that of the golden one due to the higher performance of its matching scheme (F=0.241), as can be concluded from Tables 1 and 3. Finally, one can consider the correlation degree of purely syntactic network as the lower bound, since as already pointed out; nodes in semantic network usually possess higher number of connections, hence higher correlation on degrees.

Among the *h** networks, it can be seen that *h15* network exhibits closest approximation to an ideal Web service network in terms of its S_{index} (both small world conditions are satisfied, and it grows linearly with network size), power-law exponent (holding smallest exponent degree and still showing power-law distribution) and negative correlation value on degrees. One major threat in estimating the boundaries in all three metrics in our analysis is the presence of significant occurrence of ambiguous part/element names in the given

dataset. This complicates the assumption that pure syntactic networks expose true semantic for network properties and those values can be considered for boundaries.

V. Related Work

In the light of web-service network analysis, Oh et al. [2] analyzed topological characteristics of a network constructed from small corpus of both public and artificially generated Web services descriptions. They developed a Web service benchmarking tool supporting generation of Web service descriptions such that the underlying network model is complying with those distributions and models observed in their experiment with real-world and artificially generated datasets. Cui et al. [3] utilized topological property of the Web service networks to resolve user queries over composition of services. Kil et al. [4] studied structural properties of the current web service networks and concluded that regardless of the utilized matching scheme and examined network types, all Web service networks show small world properties well and power-law like distribution to some extent. Gekas and Fasli [5] concluded that the performance of service composition algorithm is considerably influenced, among other things, by the density and link distribution of the network. While the shared observation among [3] [4][5] is that Web service networks expose characteristics of both scale-free and small world networks, the common assumption is the availability of semantically annotated web services for analysis purpose. From a different perspective, Küngas and Matskin [7] analyzed synergy between web services supplied by commercial and governmental sectors. They identified that governmental web services are more data-intensive compared to their commercial counterparts.

The closest work to ours in providing bootstrapping ontologies for Web services is presented by Segev and Sheng[18]. They combined TF/IDF measures with Web search results to discover proper domain concepts representing WSDL elements and then validate it using textual documentations in WSDL documents. Since around 94% of WSDL documents in our collection lack any textual documentation [6], the straightforward utilization of their approach is not possible for our case. However, the novelty of exploiting Web search results can be considered in our future work. Lessons learned from state of art XML schema matching solutions such as PORSCHE [19] are the potential enhancements which need to be adapted to our ontology learning module, as we are aiming at building a reference ontology rather than pair-wise schema matching solution.

VI. Conclusions and Future Work

In this paper proposed an evaluation framework for evaluating automatically semantic annotations of Web services. We used the framework to evaluate automated simple solution designed for annotating large sets of Web services from two perspectives: quality of matching and Web service network properties. We showed that the networks resulted from these annotated web services exhibit same properties as observed in a smaller scale of web service networks. We presented our analytical metrics to track the performance of adopted matching and annotation scheme based on network properties. The results of our analysis provide web-service community with grounding metrics for evaluating Web service annotation and matching schemes in settings where manual evaluation is not feasible.

We leave investigation of potential service composition solutions, which are discovered as a by-result of semantic annotation, for our future work. In addition, we intend to adopt structured data model, instead of current bag-of-word model in our ontology learning module and then evaluate this enhancement with the proposed evaluation framework.

Acknowledgment

This research is partly funded by ERDF via the Software Technology and Applications Competence Centre (STACC) and ESF via the DoRa program

References

[1] P.Küngas, and M. Dumas,"Cost-Effective Semantic Annotation of XML Schemas and Web Service Interfaces". SCC-09, pp.372-379, Sept. 2009,

[2] S.-C. Oh, D. Lee ,and S.R.T. Kumara, "Effective Web Service Composition in Diverse and Large-Scale Service Networks", *IEEE Transactions on Services Computing*, vol. 1, no. 1, pp. 15-32 , 2008,

[3] Y. Cui,S. Kumara, J. Jung-Woon Yoo ,and F. Cavdur, "Large-Scale Network Decomposition and Mathematical Programming Based Web Service Composition,",in Proc. CEC-10, pp.511-514, 2009

[4] H. Kil, S.-C. Oh, E. Elmacioglu, W. Nam, and D. Lee, "Graph theoretic topological analysis of web service networks," *Journal of World Wide Web*. vol.2, no.13,pp.321-243 , .2009,

[5] J.Gekas,and M.Fasli,"Employing Graph Theory for Web Service Composition", *IJITW*, vol.2, issue. 4, pp.21-40 , 2007

[6] S.Mokarizadeh, P.Kungas,and M.Matskin,"Ontology Learning for Cost-Effective Large-scale Semantic Annotation of XML Schemas and Web Service Interfaces".in Porc. EKAW 2010, LNAI 6317,pp.401-410, 2010

[7] P. Küngas,and M. Matskin, "Interaction and Potential Synergy between Commercial and Governmental Web Services - a Case Study". in Proc. IEEE Congress On Services, pp.1-8 , July 2007

[8] D.J Watts,and S. Strogatz,"Collective dynamics of 'small-world' networks" . *Journal of Nature*, vol. 393(4),pp:440-442 , 1998,

[9] P.Eros,and A.Renyi, "On random graphs", *Publication Mathematics*, vol. 6, pp.290-297 , 1959

[10] A. L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509-512. Oct. 1999

[11] J. Euzenat,and P. Shvaiko, *Ontology Matching*. Springer, pp.42-43,2007

[12] A.Heß, N.Kushmeric, "Machine Learning for Annotating Semantic Web services ",AAAI Spring Symposium Semantic Web Services, *2004*

[13] M.E.J. Newman ,Assortative Mixing in Networks, *Phys.Rev*.Lett.89, 208701 ,2002

[14] S.N. Kim, T.Baldwin, "Automatic Interpretation of Noun Compounds Using WordNet Similarity", *NLP*, ,Springer vol.3651 pp.945-956,2005

[15] MD.Humphries ,and K. Gurney. Network 'Small-World-Ness': A Quantitative Method for Determining Canonical Network Equivalence. *PLoS* , ONE 3(4): e0002051. 2008

[16] J. M. Montoya, and R. V. Sole, "Small world patterns in food webs". J. Theor. Biol. 214, 405–412. (doi:10.1006/jtbi. 2001.2460) , 2002

[17] A. Clauset, C. R. Shalizi, and M. E. J. Newman. "Power-law distributionsin empirical data". *Preprint, arXiv.org*:0706.1062, 2007.

[18] A. Segev and Q.Z. Sheng. Bootstraping Ontologies for Web Services, IEEE Transaction on Service Computing, Vol. 3, (To Appear), 2010

[19] K.Saleem, Z.Bellahsene, E.Hunt: PORSCHE: Performance ORiented SCHEma mediation. *Inf. Syst*. 33(7-8): 637-657 ,2008

[20] D.Lewis."Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval". ECML-98,Springer, Verlag. pp. 4–15.1998